**[PHP的SESSION机制](https://www.cnblogs.com/acpp/archive/2011/06/10/2077592.html)**

一、默认机制，用磁盘文件来实现PHP会话。php.ini配置：session.save\_handler = files

1、session\_start()

A、 session\_start()是session机制的开始，它有一定概率开启垃圾回收,因为session是存放在文件中，

PHP自身的垃圾回收是无效的，SESSION的回收是要删文件的，这个概率是根据php.ini的配置决定的，

但是有的系统是 session.gc\_probability =0，这也就是说概率是0，而是通过cron脚本来实现垃圾回收。

session.gc\_probability =1

session.gc\_divisor =1000

session.gc\_maxlifetime =1440//过期时间 默认24分钟

//概率是 session.gc\_probability/session.gc\_divisor 结果 1/1000,

//不建议设置过小，因为session的垃圾回收，是需要检查每个文件是否过期的。

session.save\_path =//好像不同的系统默认不一样，有一种设置是 "N;/path"

//这是随机分级存储，这个样的话，垃圾回收将不起作用，需要自己写脚本。

B、 session会判断当前是否有$\_COOKIE[session\_name()];session\_name()返回保存session\_id的COOKIE键值，这个值可以从php.ini找到：

session.name = PHPSESSID //默认值PHPSESSID

C、 如果不存在会生成一个session\_id,然后把生成的session\_id作为COOKIE的值传递到客户端.

相当于执行了下面COOKIE 操作，注意的是，这一步执行setcookie()操作，COOKIE是在header头中发送的，这之前是不能有输出的，PHP有另外一个函数 session\_regenerate\_id() 如果使用这个函数，这之前也是不能有输出的。

setcookie(session\_name(),

session\_id(),

session.cookie\_lifetime,//默认0

session.cookie\_path,//默认'/'当前程序跟目录下都有效

session.cookie\_domain,//默认为空

)

D、 如果存在那么session\_id =$\_COOKIE[session\_name];

然后去session.save\_path指定的文件夹里去找名字为'SESS\_'.session\_id()的文件；

读取文件的内容反序列化，然后放到$\_SESSION中；

2、 为$\_SESSION赋值

比如新添加一个值$\_SESSION['test'] ='blah'; 那么这个$\_SESSION只会维护在内存中，当脚本执行结束的时候，

用把$\_SESSION的值写入到session\_id指定的文件夹中，然后关闭相关资源. 这个阶段有可能执行更改session\_id的操作，

比如销毁一个旧的的session\_id，生成一个全新的session\_id.一半用在自定义 session操作，角色的转换上，

比如Drupal.Drupal的匿名用户有一个SESSION的，当它登录后需要换用新的session\_id

if (isset($\_COOKIE[session\_name()])) {

　　setcookie(session\_name(),'',time() -42000,'/');//旧session cookie过期

　　}

　　session\_regenerate\_id();//这一步会生成新的session\_id

　　//session\_id()返回的是新的值

3、 写入SESSION操作

在脚本结束的时候会执行SESSION写入操作，把$\_SESSION中值写入到session\_id命名的文件中，可能已经存在，

可能需要创建新的文件。

4、 销毁SESSION

SESSION发出去的COOKIE一般属于即时COOKIE，保存在内存中，当浏览器关闭后，才会过期，假如需要人为强制过期，

比如 退出登录，而不是关闭浏览器，那么就需要在代码里销毁SESSION，方法有很多，

1. setcookie(session\_name(),session\_id(),time() -8000000,..);//退出登录前执行

2. usset($\_SESSION);//这会删除所有的$\_SESSION数据，刷新后，有COOKIE传过来，但是没有数据。

3. session\_destroy();//这个作用更彻底，删除$\_SESSION 删除session文件，和session\_id

当不关闭浏览器的情况下，再次刷新，2和3都会有COOKIE传过来，但是找不到数据

二、由用户自定义session处理机制。php.ini配置：session.save\_handler = user

用户自定义session处理机制，更加直观

\*session\_set\_save\_handler('open','close','read','write','destroy','gc');

1.session\_start(),

执行open($save\_path,$session\_name)打开session操作句柄

$save\_path 在session.save\_handler = files的情况下它就是session.save\_path，

但是如果用户自定的话，这个两个参数都用不上，直接返回TRUE

执行read($id)从中读取数据.//这个参数是自动传递的就是session\_id(),可以通过这个值进行操作。

\*2.脚本执行结束

执行write($id,$sess\_data) //两个参数，很简单

\*3.假如用户需要session\_destroy()

先执行destroy.在执行第2步